

APPARATUS AND METHOD FOR TRACE STREAM  
IDENTIFICATION OF A PROCESSOR RESET

This application claims priority under 35 USC §119(e)(1) of Provisional Application Number 60/434,105 (TI-34663P) filed December 17, 2002.

**Related Applications**

- 5 U.S. Patent Application (Attorney Docket No. TI-34654),  
entitled APPARATUS AND METHOD FOR SYNCHRONIZATION OF TRACE  
STREAMS FROM MULTIPLE PROCESSING UNITS, invented by Gary L.  
Swoboda, filed on even date herewith, and assigned to the  
assignee of the present application; U.S. Patent  
10 Application (Attorney Docket No. TI-34655), entitled  
APPARATUS AND METHOD FOR SEPARATING DETECTION AND ASSERTION  
OF A TRIGGER EVENT, invented by Gary L. Swoboda, filed on  
even date herewith, and assigned to the assignee of the

5 present application; U.S. Patent Application (Attorney  
Docket No. TI-34656), entitled APPARATUS AND METHOD FOR  
STATE SELECTABLE TRACE STREAM GENERATION, invented by Gary  
L. Swoboda, filed on even date herewith, and assigned to  
the assignee of the present application; U.S. Patent  
10 Application (Attorney Docket No. TI-34657), entitled  
APPARATUS AND METHOD FOR SELECTING PROGRAM HALTS IN AN  
UNPROTECTED PIPELINE AT NON-INTERRUPTIBLE POINTS IN CODE  
EXECUTION, invented by Gary L. Swoboda, filed on even date  
herewith, and assigned to the assignee of the present  
15 application; U.S. Patent Application (Attorney Docket No.  
TI-34658), entitled APPARATUS AND METHOD FOR REPORTING  
PROGRAM HALTS IN AN UNPROTECTED PIPELINE AT NON-  
INTERRUPTIBLE POINTS IN CODE EXECUTION, invented by Gary L.  
Swoboda, filed on even date herewith, and assigned to the  
20 assignee of the present application; U.S. Patent  
Application (Attorney Docket No. TI-34659), entitled  
APPARATUS AND METHOD FOR A FLUSH PROCEDURE IN AN  
INTERRUPTED TRACE STREAM, invented by Gary L. Swoboda,  
filed on even date herewith, and assigned to the assignee  
25 of the present application; U.S. Patent Application  
(Attorney Docket No. TI-34660), entitled APPARATUS AND  
METHOD FOR CAPTURING AN EVENT OR COMBINATION OF EVENTS  
RESULTING IN A TRIGGER SIGNAL IN A TARGET PROCESSOR,  
invented by Gary L. Swoboda, filed on even date herewith,  
30 and assigned to the assignee of the present application;  
U.S. Patent Application (Attorney Docket No. TI-34661),

5 entitled APPARATUS AND METHOD FOR CAPTURING THE PROGRAM  
COUNTER ADDRESS ASSOCIATED WITH A TRIGGER SIGNAL IN A  
TARGET PROCESSOR, invented by Gary L. Swoboda, filed on  
even date herewith, and assigned to the assignee of the  
present application; U.S. Patent Application (Attorney  
10 Docket No. TI-34662), entitled APPARATUS AND METHOD  
DETECTING ADDRESS CHARACTERISTICS FOR USE WITH A TRIGGER  
GENERATION UNIT IN A TARGET PROCESSOR, invented by Gary L.  
Swoboda and Jason L. Peck, filed on even date herewith, and  
assigned to the assignee of the present application; U.S.  
15 Patent (Attorney Docket No. TI-34664), entitled APPARATUS  
AND METHOD FOR TRACE STREAM IDENTIFICATION OF A PROCESSOR  
DEBUG HALT SIGNAL, invented by Gary L. Swoboda and Bryan  
Thome, filed on even date herewith, and assigned to the  
assignee of the present application; U.S. Patent  
20 Application (Attorney Docket No. TI-34665), entitled  
APPARATUS AND METHOD FOR TRACE STREAM IDENTIFICATION OF A  
PIPELINE FLATTENER PRIMARY CODE FLUSH FOLLOWING INITIATION  
OF AN INTERRUPT SERVICE ROUTINE; invented by Gary L.  
Swoboda and Bryan Thome, filed on even date herewith, and  
25 assigned to the assignee of the present application; U.S.  
Patent Application (Attorney Docket No. TI-34666), entitled  
APPARATUS AND METHOD FOR TRACE STREAM IDENTIFICATION OF A  
PIPELINE FLATTENER SECONDARY CODE FLUSH FOLLOWING A RETURN  
TO PRIMARY CODE EXECUTION, invented by Gary L. Swoboda and  
30 Bryan Thome, filed on even date herewith, and assigned to  
the assignee of the present application; U.S. Patent

5 Application (Docket No. TI-34667), entitled APPARATUS AND  
METHOD IDENTIFICATION OF A PRIMARY CODE START SYNC POINT  
FOLLOWING A RETURN TO PRIMARY CODE EXECUTION, invented by  
Gary L. Swoboda, filed on even date herewith, and assigned  
to the assignee of the present application; U. S. Patent  
10 Application (Attorney Docket No. TI-34668), entitled  
APPARATUS AND METHOD FOR IDENTIFICATION OF A NEW SECONDARY  
CODE START POINT FOLLOWING A RETURN FROM A SECONDARY CODE  
EXECUTION, invented by Gary L. Swoboda, filed on even date  
herewith, and assigned to the assignee of the present  
15 application; U.S. Patent Application (Attorney Docket No.  
TI-34669), entitled APPARATUS AND METHOD FOR TRACE STREAM  
IDENTIFICATION OF A PAUSE POINT IN A CODE EXECUTION  
SEQUENCE, invented by Gary L. Swoboda, filed on even date  
herewith, and assigned to the assignee of the present  
20 application; U.S. Patent Application (Attorney Docket No.  
TI-34670), entitled APPARATUS AND METHOD FOR COMPRESSION OF  
A TIMING TRACE STREAM, invented by Gary L. Swoboda and  
Bryan Thome, filed on even date herewith, and assigned to  
the assignee of the present application; U.S. Patent  
25 Application (Attorney Docket No. TI-34671), entitled  
APPARATUS AND METHOD FOR TRACE STREAM IDENTIFICATION OF  
MULTIPLE TARGET PROCESSOR EVENTS, invented by Gary L.  
Swoboda and Bryan Thome, filed on even date herewith, and  
assigned to the assignee of the present application; and  
30 U.S. Patent Application (Attorney Docket No. TI-34672),  
entitled APPARATUS AND METHOD FOR OP CODE EXTENSION IN

5    PACKET GROUPS TRANSMITTED IN TRACE STREAMS, invented by  
Gary L. Swoboda and Bryan Thome, filed on even date  
herewith, and assigned to the assignee of the present  
application are related applications.

10    **Background of the Invention**

1.    Field of the Invention

This invention relates generally to the testing of digital  
15    signal processing units and, more particularly, to the  
signals that are transmitted from a target processor to a  
host processing to permit analysis of the target processing  
unit operation.    Certain events in the target processor  
must be communicated to the host processing unit along with  
20    contextual information.    In this manner, the test and debug  
data can be analyzed and problems in the operation of the  
target processor identified.

2.    Description of the Related Art

25    As microprocessors and digital signal processors have  
become increasingly complex, advanced techniques have been  
developed to test these devices.    Dedicated apparatus is  
available to implement the advanced techniques.    Referring  
30    to Fig. 1A, a general configuration for the test and debug  
of a target processor **12** is shown.    The test and debug

5 procedures operate under control of a host processing unit  
10. The host processing unit 10 applies control signals to  
the emulation unit 11 and receives (test) data signals from  
the emulation unit 11 by cable connector 14. The emulation  
unit 11 applies control signals to and receives (test)  
10 signals from the target processing unit 12 by connector  
cable 15. The emulation unit 11 can be thought of as an  
interface unit between the host processing unit 10 and the  
target processor 12. The emulation unit 11 processes the  
control signals from the host processor unit 10 and applies  
15 these signals to the target processor 12 in such a manner  
that the target processor will respond with the appropriate  
test signals. The test signals from the target processor  
12 can be a variety types. Two of the most popular test  
signal types are the JTAG (Joint Test Action Group) signals  
20 and trace signals. The JTAG protocol provides a  
standardized test procedure in wide use in which the status  
of selected components is determined in response to control  
signals from the host processing unit. Trace signals are  
signals from a multiplicity of selected locations in the  
25 target processor 12 during defined period of operation.  
While the width of the bus 15 interfacing to the host  
processing unit 10 generally has a standardized dimension,  
the bus between the emulation unit 11 and the target  
processor 12 can be increased to accommodate an increasing  
30 amount of data needed to verify the operation of the target  
processing unit 12. Part of the interface function between

5 the host processing unit **10** and the target processor **12** is to store the test signals until the signals can be transmitted to the host processing unit **10**.

Referring to Fig. 1B, the operation of the trigger  
10 generation unit **19** is shown. The trigger unit provides the main component by which the operation/state of the target processor can be altered. At least one event signal is applied to the trigger generation unit **19**. Based on the identity of the event signal(s) applied to the trigger  
15 generation unit **19**, a trigger signal is selected. Certain events and combination of events, referred to as an event front, generate a selected trigger signal that results in certain activity in the target processor such as a debug halt. Combinations of different events generating trigger  
20 signals are referred to as jobs. Multiple jobs can have the same trigger signal or combination of trigger signals. In the test and debug of the target processor, the trigger signals can provide impetus for changing state in the target processor or for performing a specified activity.  
25 The event front defines the reason for the generation of trigger signal. This information is important in understanding the operation of the target processor because, as pointed out above, several combinations of events can result in the generation of a trigger signal.  
30 In order to analyze the operation of the target processing unit, the portion of the code resulting in the trigger

5 signal must be identified. However, the events in the host  
processor leading to the generation of event signals can be  
complicated. Specifically, the characteristics of an  
instruction at a program counter address can determine  
whether a trigger signal should be generated. A trigger  
10 signal can be an indication of when an address is within a  
range of addresses, outside of a range of addresses, some  
combination of address characteristics, and/or the address  
is aligned with a reference address. In this instance, the  
address can be the program address of an instruction or a  
15 memory address directly or indirectly referenced by a  
program instruction.

In testing the target processors, certain events must be  
identified. The reset procedure is a normal system  
20 function in which the (target) processor halts the current  
processing activity and begins a processing activity at  
preselected program point. However, it is important to  
determine the point in the original processing procedure  
where the reset procedure occurs, the duration of the reset  
25 procedure, and the starting point of the processing  
procedure. A reset of a processor can be initiated by the  
user or, in the processor itself, by identification of a  
preselected condition.

30 A need has been felt for apparatus and an associated method  
having the feature that a reset procedure is identified in



5 a target processor and occurrence of the reset procedure  
communicated to the testing apparatus. It is another  
feature of the apparatus and associated method to transfer  
information concerning a reset procedure to the testing  
processing unit using the trace stream. It is a still  
10 further feature of the apparatus and associated method to  
communicate to the testing apparatus when the reset feature  
is initiated during the program execution. It is yet  
another feature of the present in invention to communicate  
to the testing processing unit the duration of the reset  
15 procedure.

#### **Summary of the Invention**

The aforementioned and other features are accomplished,  
20 according to the present invention, by providing the target  
processor with at least two trace streams. One of the  
trace streams is a timing trace stream. The second trace  
stream, when a reset signal resulting from a user input or  
selected target processor parameters is identified, is  
25 provided with a sync marker. The sync marker includes at  
least one portion identifying the event as a reset  
procedure, a portion relating the reset procedure to the  
timing trace stream, and a portion identifying the point in  
the program execution when the reset procedure is  
30 initiated. In addition, the second trace stream includes a  
sync marker including at least one packet identifying when

5 the reset procedure is ended. In the preferred embodiment,  
the second trace stream is a program counter trace stream.  
The point in the program execution where the reset signal  
is identified is determined by the program counter address.  
The time of the occurrence of the reset procedure is  
10 determined by trace synchronization markers and by a  
position of a clock cycle in a timing packet. The end of  
the reset procedure is marked by a sync marker that  
includes a header identifying the event as an end of reset  
and relating the event to the timing trace stream.

15

Other features and advantages of present invention will be  
more clearly understood upon reading of the following  
description and the accompanying drawings and the claims.

## 5    **Brief Description of the Drawings**

Figure 1A is a general block diagram of a system configuration for test and debug of a target processor, while Fig. 1B illustrates the function of the trigger unit.

10

Figure 2 is a block diagram of selected components in the target processor used the testing of the central processing unit of the target processor according to the present invention.

15

Figure 3 is a block diagram of selected components of the illustrating the relationship between the components transmitting trace streams in the target processor.

20    Figure 4A illustrates format by which the timing packets are assembled according to the present invention, while Figure 4B illustrates the inclusion of a periodic sync marker in the timing trace stream according to the present invention.

25

Figure 5 illustrates the parameters for sync markers in the program counter stream packets according to the present invention.

30    Figure 6A illustrates the sync markers in the program counter trace stream when a periodic sync point ID is

5 generated, while Figure 6B illustrates the reconstruction of the target processor operation from the trace streams according to the present invention.

Figure 7A is a block diagram illustrating the apparatus  
10 used in reconstructing the processor operation from the trace streams according to the present invention, while Figure 7B is block diagram illustrating where the program counter identification of instructions is provided for the trace streams according to the present invention.

15 Figure 8A is block diagram of the program counter sync marker generator unit; Figure 8B illustrates the sync markers generated in the presence of a debug halt condition; and Figure 8C illustrates the reconstruction of  
20 the processor operation from the trace stream according to the present invention.

### **Description of the Preferred Embodiment**

#### **25 1. Detailed Description of the Figures**

Fig. 1A and Fig. 1B have been described with respect to the related art.

30 Referring to Fig. 2, a block diagram of selected components of a target processor **20**, according to the present

5 invention, is shown. The target processor includes at least one central processing unit **200** and a memory unit **208**. The central processing unit **200** and the memory unit **208** are the components being tested. The trace system for testing the central processing unit **200** and the memory unit

10 **202** includes three packet generating units, a data packet generation unit **201**, a program counter packet generation unit **202** and a timing packet generation unit **203**. The data packet generation unit **201** receives VALID signals, READ/WRITE signals and DATA signals from the central

15 processing unit **200**. After placing the signals in packets, the packets are applied to the scheduler/multiplexer unit **204** and forwarded to the test and debug port **205** for transfer to the emulation unit **11**. The program counter packet generation unit **202** receives PROGRAM COUNTER

20 signals, VALID signals, BRANCH signals, and BRANCH TYPE signals from the central processing unit **200** and, after forming these signal into packets, applies the resulting program counter packets to the scheduler/multiplexer **204** for transfer to the test and debug port **205**. The timing

25 packet generation unit **203** receives ADVANCE signals, VALID signals and CLOCK signals from the central processing unit **200** and, after forming these signal into packets, applies the resulting packets to the scheduler/multiplexer unit **204** and the scheduler/multiplexer **204** applies the packets to

30 the test and debug port **205**. Trigger unit **209** receives EVENT signals from the central processing unit **200** and

5 signals that are applied to the data trace generation unit  
201, the program counter trace generation unit 202, and the  
timing trace generation unit 203. The trigger unit 209  
applies TRIGGER and CONTROL signals to the central  
processing unit 200 and applies CONTROL (i.e., STOP and  
10 START) signals to the data trace generation unit 201, the  
program counter generation unit 202, and the timing trace  
generation unit 203. The sync ID generation unit 207  
applies signals to the data trace generation unit 201, the  
program counter trace generation unit 202 and the timing  
15 trace generation unit 203. While the test and debug  
apparatus components are shown as being separate from the  
central processing unit 201, it will be clear that an  
implementation these components can be integrated with the  
components of the central processing unit 201.

20

Referring to Fig. 3, the relationship between selected  
components in the target processor 20 is illustrated. The  
data trace generation unit 201 includes a packet assembly  
unit 2011 and a FIFO (first in/first out) storage unit  
25 2012, the program counter trace generation unit 202  
includes a packet assembly unit 2021 and a FIFO storage  
unit 2022, and the timing trace generation unit 203  
includes a packet generation unit 2031 and a FIFO storage  
unit 2032. As the signals are applied to the packet  
30 generators 201, 202, and 203, the signals are assembled  
into packets of information. The packets in the preferred

embodiment are 10 bits in width. Packets are assembled in the packet assembly units in response to input signals and transferred to the associated FIFO unit. The scheduler/multiplexer **204** generates a signal to a selected trace generation unit and the contents of the associated FIFO storage unit are transferred to the scheduler/multiplexer **204** for transfer to the emulation unit. Also illustrated in Fig. 3 is the sync ID generation unit **207**. The sync ID generation unit **207** applies an SYNC ID signal to the packet assembly unit of each trace generation unit. The periodic signal, a counter signal in the preferred embodiment, is included in a current packet and transferred to the associated FIFO unit. The packet resulting from the SYNC ID signal in each trace is transferred to the emulation unit and then to the host processing unit. In the host processing unit, the same count in each trace stream indicates that the point at which the trace streams are synchronized. In addition, the packet assembly unit **2031** of the timing trace generation unit **203** applies an INDEX signal to the packet assembly unit **2021** of the program counter trace generation unit **202**. The function of the INDEX signal will be described below.

Referring to Fig. 4A, the assembly of timing packets is illustrated. The signals applied to the timing trace generation unit **203** are the CLOCK signals and the ADVANCE signals. The CLOCK signals are system clock signals to

5 which the operation of the central processing unit **200** is  
synchronized. The ADVANCE signals indicate an activity  
such as a pipeline advance or program counter advance (( ))  
or a pipeline non-advance or program counter non-advance  
(1). An ADVANCE or NON-ADVANCE signal occurs each clock  
10 cycle. The timing packet is assembled so that the logic  
signal indicating ADVANCE or NON-ADVANCE is transmitted at  
the position of the concurrent CLOCK signal. These  
combined CLOCK/ADVANCE signals are divided into groups of 8  
signals, assembled with two control bits in the packet  
15 assembly unit **2031**, and transferred to the FIFO storage  
unit **2032**.

Referring to Fig. 4B, the trace stream generated by the  
timing trace generation unit **203** is illustrated. The first  
20 (in time) trace packet is generated as before. During the  
assembly of the second trace packet, a SYYN ID signal is  
generated during the third clock cycle. In response, the  
timing packet assembly unit **2031** assembles a packet in  
response to the SYNC ID signal that includes the sync ID  
25 number. The next timing packet is only partially assembled  
at the time of the SYNC ID signal. In fact, the SYNC ID  
signal occurs during the third clock cycle of the formation  
of this timing packet. The timing packet assembly unit  
**2031** generates a TIMING INDEX 3 signal (for the third  
30 packet clock cycle at which the SYNC ID signal occurs) and



5 transmits this TIMING INDEX 3 signal to the program counter packet assembly unit **2031**.

Referring to Fig. 5, the parameters of a sync marker in the program counter trace stream, according to the present invention is shown. The program counter stream sync markers each have a plurality of packets associated therewith. The packets of each sync marker can transmit a plurality of parameters. A SYNC POINT TYPE parameter defines the event described by the contents of the accompanying packets. A program counter TYPE FAMILY parameter provides a context for the SYNC POINT TYPE parameter and is described by the first two most significant bits of a second header packet. A BRANCH INDEX parameter in all but the final SYNC POINT points to a bit within the next relative branch packet following the SYNC POINT. When the program counter trace stream is disabled, this index points a bit in the previous relative branch packet when the BRANCH INDEX parameter is not a logic "0". In this situation, the branch register will not be complete and will be considered as flushed. When the BRANCH INDEX is a logic "0", this value point to the least significant value of branch register and is the oldest branch in the packet. A SYNC ID parameter matches the SYNC POINT with the corresponding TIMING and/or DATA SYNC POINT which are tagged with the same SYNC ID parameter. A TIMING INDEX parameter is applied relative to a corresponding TIMING

5 SYNC POINT. For all but LAST POINT SYNC events, the first timing packet after the TIMING PACKET contains timing bits during which the SYNC POINT occurred. When the timing stream is disabled, the TIMING INDEX points to a bit in the timing packet just previous to the TIMING SYNC POINT packet  
10 when the TIMING INDEX value is not zero. In this situation, the timing packet is considered as flushed. A TYPE DATA parameter is defined by each SYNC TYPE. An ABSOLUTE PC VALUE is the program counter address at which the program counter trace stream and the timing information  
15 are aligned. An OFFSET COUNT parameter is the program counter offset counter at which the program counter and the timing information are aligned.

Referring to Fig. 6A, a program counter trace stream for a  
20 hypothetical program execution is illustrated. In this program example, the execution proceeds without interruption from external events. The program counter trace stream will consist of a first sync point marker 601, a plurality of periodic sync point ID markers 602, and last  
25 sync point marker 603 designating the end of the test procedure. The principal parameters of each of the packets are a sync point type, a sync point ID, a timing index, and an absolute PC value. The first and last sync points identify the beginning and the end of the trace stream.  
30 The sync ID parameter is the value from the value from the most recent sync point ID generator unit. In the preferred

5   embodiment, this value in a 3-bit logic sequence. The  
timing index identifies the status of the clock signals in  
a packet, i.e., the position in the 8 position timing  
packet when the event producing the sync signal occurs.  
And the absolute address of the program counter at the time  
10   that the event causing the sync packet is provided. Based  
on this information, the events in the target processor can  
be reconstructed by the host processor.

Referring to Fig. 6B, the reconstruction of the program  
15   execution from the timing and program counter trace streams  
is illustrated. The timing trace stream consists of  
packets of 8 logic "0"s and logic "1"s. The logic "0"s  
indicate that either the program counter or the pipeline is  
advanced, while the logic "1"s indicate the either the  
20   program counter or the pipeline is stalled during that  
clock cycle. Because each program counter trace packet has  
an absolute address parameter, a sync ID, and the timing  
index in addition to the packet identifying parameter, the  
program counter addresses can be identified with a  
25   particular clock cycle. Similarly, the periodic sync  
points can be specifically identified with a clock cycle in  
the timing trace stream. In this illustration, the timing  
trace stream and the sync ID generating unit are in  
operation when the program counter trace stream is  
30   initiated. The periodic sync point is illustrative of the  
plurality of periodic sync points that would typically be

5 available between the first and the last trace point, the periodic sync points permitting the synchronization of the three trace streams for a processing unit.

Referring to Fig. 7A, the general technique for  
10 reconstruction of the trace streams is illustrated. The trace streams originate in the target processor **12** as the target processor **12** is executing a program **1201**. The trace signals are applied to the host processing unit **10**. The host processing unit **10** also includes the same program  
15 **1201**. Therefore, in the illustrative example of Fig. 6 wherein the program execution proceeds without interruptions or changes, only the first and the final absolute addresses of the program counter are needed. Using the advance/non-advance signals of the timing trace  
20 stream, the host processing unit can reconstruct the program as a function of clock cycle. Therefore, without the sync ID packets, only the first and last sync markers are needed for the trace stream. This technique results in reduced information transfer. Fig. 6 includes the presence  
25 of periodic sync ID cycles, of which only one is shown. The periodic sync ID packets are important for synchronizing the plurality of trace streams, for selection of a particular portion of the program to analyze, and for restarting a program execution analysis for a situation  
30 wherein at least a portion of the data in the trace data stream is lost. The host processor can discard the

5 (incomplete) trace data information between two sync ID packets and proceed with the analysis of the program outside of the sync timing packets defining the lost data.

As indicated in Fig. 6A, the program counter trace stream  
10 includes the absolute address of the program counter for an instruction. Referring to Fig. 7B, each processor includes a processor pipeline **71**. When the instruction leaves the processor pipeline, the instruction is entered in the pipeline flattener **73**. At the same time, an access of  
15 memory unit **72** is performed. The results of the memory access of memory unit **72**, which may take several clock cycles, is then merged the associated instruction in the pipeline flattener **73** and withdrawn from the pipeline flattener **73** for appropriate distribution. The pipeline  
20 flattener **73** provides a technique for maintaining the order of instructions while providing for the delay of a memory access. In the preferred embodiment, the absolute address used in the program counter trace stream is the derived from the instruction of leaving the pipeline flattener **71**.  
25 As a practical matter, the absolute address is delayed. It is not necessary to use a pipeline flattener **73**. The instructions can have appropriate labels associated therewith to eliminate the need for the pipeline flattener **73**.

30

5 Referring to Fig. 8A, the major components of the program counter packet generation unit **202** is shown. The program counter packet generation unit **202** includes a decoder unit **2023**, storage unit **2021**, a FIFO unit **2022**, and a gate unit **2024**. PERIODIC SYNC ID signals, TIMING INDEX signals, and  
10 ABSOLUTE ADDRESS signals are applied to gate unit **2024**. When the PERIODIC SYNC ID signals are incremented, the decoder unit **2023**, in response to the PERIODIC SYNC ID signal, stores a periodic sync ID header signal group in a predetermined location **2021A** of the header portion of the  
15 storage unit **2021**. The PERIODIC SYNC signal causes the gate **2024** to transmit the PERIODIC SYNC ID signals, the TIMING INDEX signals and the ABSOLUTE ADDRESS signals. These transmitted signals are stored in the storage unit **2021** in information packet locations assigned to these  
20 parameters. When all of the portions of the periodic sync marker have been assembled in the storage unit **2021**, then the component packets of the periodic sync marker are transferred to the FIFO unit **2022** for eventual transmission to the scheduler/multiplexer unit. Similarly, when a RESET  
25 signal is generated and applied to the decoder unit **2023**, the reset header identifying signal group is stored in position **2021A** in the header portion of the storage unit **2021**. The RESET signal applied to decoder unit **2023** results in a control signal being applied to the gate **2024**.  
30 As a result of the control signal, the SYNC ID signals, the TIMING INDEX signals, and the ABSOLUTE ADDRESS signals are

5 stored in the appropriate locations in storage unit **2021**.  
When the reset signal sync marker has been assembled, i.e.,  
in packets, the sync marker is transferred to the FIFO unit  
**2022**. When the RESET signal is removed, decoder unit **2023**  
causes reset-off identifying signals to be stored in the  
10 header location **2021A**. The output signal from the decoder  
unit **2023** As a result of the RESET signal being removed  
from gate **2023**, a control signal is applied to gate **2024**  
resulting in the storage of the SYNC ID signals, the TIMING  
INDEX signals, and the ABSOLUTE ADDRESS signals storage  
15 unit **2021** prior to transfer of the sync marker packets to  
the FIFO unit **2022**. As will be clear, the first  
(instruction) sync point marker and the last sync point  
marker are formed in an analogous manner.

20 Referring to Fig. 8B, examples of the sync markers in the  
program counter trace stream are shown. The start of the  
test procedure is shown in first point sync marker 801.  
Thereafter, periodic sync ID markers 805 can be generated.  
Other event markers can also be generated. The  
25 identification of a RESET signal results in the generation  
of reset sync marker 810. When the RESET signal is  
removed, a reset-off sync marker 820 is generated. Between  
the reset sync marker 810 and the reset-off sync marker  
815, one or more periodic sync ID signals can be generated,  
30 the number depending on the time interval between the reset  
sync marker 810 and the reset-off sync marker 815.

5

Referring to Fig. 8C, the reconstruction of the program counter trace stream from the sync markers of Fig. 8B and the timing trace stream is shown. The first sync point marker indicates the beginning of test procedure with a program counter address of PC. The program continues to execute unit with the program counter addresses being related to a particular processor clock cycle. When the RESET signal is generated, the program counter is at address  $PC + N+2$  and is related to a particular clock cycle. Thereafter, the program counter does not advance as indicated by the logic "1"s associated with each clock cycle. When the RESET signal is removed, the reset-off marker is generated. The program counter is at address PC' and the target processor begins processing at a preselected absolute address.

20

## 2. Operation of the Preferred Embodiment

The present invention relies on the ability to relate the timing trace stream and the program counter trace stream. This relationship is provided by having periodic sync ID information transmitted in each trace stream. In addition, the timing packets are grouped in packets of eight signals identifying whether the program counter (or the pipeline address) advanced or didn't advance. The sync markers in the program counter stream include both the periodic sync

30



5 ID information and an index indicating the position in the  
current eight-position timing packet when the event  
occurred. Thus, the clock cycle of the event can be  
specified. In addition, the address of the program counter  
is provided in the program counter sync markers so that the  
10 event can be related to the execution of the program. As a  
result, when a reset sync signal is generated, the location  
of the signal relative to the target processor clock and to  
the program execution is established and can be  
reconstructed. In the preferred embodiment, data (memory  
15 access) information is transferred from the target  
processor to the host processing unit. It is therefore  
possible to reconstruct the entire operation of the target  
processor relative to the reset signal from the transmitted  
trace streams.

20

The sync marker trace streams illustrated above relate to an  
idealized operation of the target processor in order to  
emphasize the features of the present invention. Numerous  
other sync events (e.g. branch events) will typically be  
25 entered in the storage unit and included in the program  
counter trace stream.

In the foregoing discussion, the sync markers can have  
additional information embedded therein depending on the  
30 implementation of the apparatus generating and interpreting  
the trace streams. This information will be related to the

5 parameters shown in Fig. 5. It will also be clear that a  
data trace stream, as shown in Fig. 2 will typically be  
present. The periodic sync IDs as well as the timing  
indexes will also be included in the data trace stream. In  
addition, the program counter absolute address parameter  
10 can be replaced by the program counter off-set register in  
certain situations.

While the invention has been described with respect to the  
embodiments set forth above, the invention is not  
15 necessarily limited to these embodiments. Accordingly,  
other embodiments, variations, and improvements not  
described herein are not necessarily excluded from the  
scope of the invention, the scope of the invention being  
defined by the following claims.